

LISTING OF THE CLAIMS:

1. (Currently Amended) A method, comprised of enhancing a computational service to each client of a plurality of clients, by:

moving a selected portion of a computation from a server into a trusted co-server executing inside a secure coprocessor; and

allowing each client to interact with the server and the co-server; and

using the trusted co-server as a trusted third party in interactions between the client and the server.

2. (Original) A method as in Claim 1, wherein the step of allowing includes providing a trusted third party at said server.

3. (Original) A method as recited in Claim 1, wherein said step of allowing includes enabling said client an authenticated, private channel to said co-server.

4. (Original) A method as in Claim 1, wherein said service is a Web service and said clients are remote users operating browsers.

5. (Original) A method as in Claim 3, wherein said step of enabling includes the client using the co-server's certified keypair to establish a shared symmetric key.

6. (Original) A method as in Claim 5, wherein said step of enabling includes employing the Secure Sockets Layer (SSL) protocol.

7. (Original) A method as in Claim 1, wherein said step of moving includes integrating functions of said co-server in a same machine as said server.

8. (Original) A method as in Claim 1, wherein said step of enhancing includes providing a desired security and/or privacy property.

9. (Original) A method as in Claim 1, wherein said step of enhancing includes providing at least one security and/or privacy property to an application selected from the group including: authentication of clients, nonrepudiation of client activity, nonrepudiation of server activity, credit card transaction security, taxes on e-commerce activity, re-selling of intellectual property, privacy of sensitive or proprietary web activity, correctness of web activity, enforcement of logo and/or "seal of approval" licenses, safety of downloadable content, authenticity of downloadable content, integrity of server machine, and any combination of these.

10. (Currently Amended) A method as in Claim 1, wherein:
input from said client is prompt from server for the user's private authenticator data, ~~such as a~~ password, input from said server is this authentication data, co-server algorithm that generates output to ~~said a~~ client based on said current co-server state and said inputs indicates whether or not the authenticator data is correct for this user.

11. (Currently Amended) A method as in Claim 1, where co-server algorithm that generates output to said server based on said a current co-server state and said inputs includes a signed statement, using a private key known to the co-server, attesting, for the server, that the client engaged in an interaction satisfying certain properties.

12. (Currently Amended) A method as in Claim 1, where co-server algorithm that generates output to said client based on said a current co-server state and said inputs includes a signed statement, using a privacy key known to the co-server, attesting, for the client, that the server engaged in an interaction satisfying certain properties.

13. (Currently Amended) A method as in Claim 1, wherein:
the client's input includes a credit card number (CCN), the output co-server algorithm that generates output to said client based on said a current co-server state and said inputs includes the CCN, encrypted so that the server cannot read it but an acquirer can.

14. (Currently Amended) A method as in Claim 13, wherein:
the server's input includes a transaction amount, the output co-server algorithm that generates output to said client based on said a current co-server state and said inputs includes the transaction amount, cryptographically bound to the encrypted CCN so that the server cannot alter it.

15. (Original) A method as in Claim 1, where:

the client's input includes a credit card number,

the server's input includes a transaction amount,

the co-server encrypts this CCN so that the server cannot read it but an acquirer can, and cryptographically binds the transaction amount to the this encrypted CCN, then, at some point during or after the interaction, transmits this data to the acquirer in such a manner so that the acquirer can receive this transaction exactly once.

16. (Currently Amended) A method as in Claim 1, wherein:

the interaction via the server input and/or the client input, includes a transaction amount A, the co-server input may include an accumulated total, the function co-server algorithm that generates new co-server state based on ~~said a~~ current co-server state and said inputs updates the accumulated amount by adding T(A), where T is a predefined function, ~~such as: a map from an amount to the taxes owed on that amount;~~ and at some point during or after this interaction, the co-server produces an authenticated statement of the current value of the accumulated amount.

17. (Currently Amended) A method as in Claim 1, where:

a remote party is an owner of intellectual property,

the server input includes part of this property, encrypted so that only the co-server can decrypt it,

the output function co-server algorithm that generates output to said client based on ~~said a~~ current co-server state and said inputs to the client includes a portion of the a decryption of input from said client.

18. (Original) A method as in Claim 17, except the output function co-server algorithm that generates output to said client based on said current co-server state and said inputs now includes a transformation of a portion of the decryption of input from said server, where said transformation may include adding a watermark.

19. (Original) A method as in Claim 17, except the output function now includes a transformation of a portion of the decryption of input from said server, where said transformation may include reducing the quality of the plaintext.

20. (Original) A method as in Claim 17, except the output function now includes a portion of the decryption of input from said server, re-encrypted, possibly with rights management rules, in a manner that a secure coprocessor at the client site can decrypt it.

21. (Original) A method as in Claim 1, wherein:

the client input includes a choice of which record R in a set of records the client would like to receive, the co-server includes this record R in its response to the client, however, the co-server obtains R in such a way as the server does not know which record was the one selected.

22. (Currently Amended) A method as in Claim 1, wherein:

a remote party establishes a content evaluation scheme, consisting of an evaluation function mapping content to some set of indicators, and as part of computing the client output function co-server algorithm that generates output to said client based on said a current co-server state and said

inputs, the co-server calculates, or verifies an external calculation, of the evaluation function and includes the result in the client output.

23. (Original) method as in Claim 22, where the evaluation function consists of determining whether specified server input from specified server merits a logo or seal of approval, in accordance with a business arrangement between the server and the remote party.

24. (Original) A method as in Claim 22, where the evaluation function consists of determining whether server input which has potentially executable content is free of viruses.

25. (Original) A method as in Claim 24, where the evaluation function is parameterized by a "signature file" and where the client output includes an identification of which signature file was used in this interaction.

26. (Original) A method as in Claim 22, where party the remote party has injected evaluation function and/or some of its parameters into the co-server through a private channel, so that the server cannot know the details of the evaluation function execution occurring on the co-server.

27. (Original) A method as in Claim 22, where the server input includes both content and a signature on the content, from one of possibly many content providers, and the evaluation function includes testing whether the signature is valid.

28. (Currently Amended) A method as in Claim 1, where:

a remote party establishes a content evaluation scheme, consisting of an evaluation function mapping content to some set of indicators, and as part of computing the server output function co-server algorithm that generates output to said client based on said a current co-server state and said inputs or internal function co-server algorithm that generates new co-server state based on said current co-server state and said inputs the co-server calculates, or verifies an external calculation, of the evaluation function input from said client and includes the result in the output.

29. (Currently Amended) A method as in Claim 1, where:

the co-server has the ability to carry out security-enhancing actions against the server, ~~such as booting the server and securely or carrying out a security scan of the server,~~ and the output returned to client indicates which of these actions have been carried out, and how recently.

30. (Currently Amended) A method as in Claim 1, where:

the client can specify whether the interaction
is a read interaction or a write interaction;
for a write interaction:

the client input includes a message M and a specification S of the appropriate entities who can read this message; the co-server retains M and S by storing them in some combination across the co-server and server via an algorithm that generates new co-server state based on said current co-server state and said inputs, the internal state in the co-server and co-server algorithm that generates output to said server based on said a current co-server state and said inputs;

however in said write interaction:

any portion of M sent via co-server algorithm that generates output to said server based on said current co-server state and said inputs is encrypted, so that the server cannot access the plaintext; and mechanisms are used to ensure that, when the co-server later retrieves any of this data from the server, that the data has not been changed;

for a read interaction:

the client input specifies which message M the client would like to read, the co-server retrieves S; if the client satisfies S, then the co-server sends M back to the client, after first retrieving and decrypting it, if necessary.

31. (Currently Amended) A method for enhancing a service to provide security and/or privacy to each client from a plurality of clients, said service including computation on a server controlled by an operator, the method comprising:

moving a selected portion of said computation from a server controlled by said operator into a trusted co-server executing inside a secure coprocessor; and

allowing clients to interact with the server through the co-server; and

using the trusted co-server as a trusted third party in interactions between the client and the server.

32. (Original) A method as recited in Claim 31 wherein the secure coprocessor is installed at the server.

33. (Currently Amended) A method for enhancing a service including computation on a server controlled by an operator, the method comprising:

providing at least one security and privacy property to at least one client from a plurality of clients by:

moving a selected portion of said computation from a server controlled by said operator into a trusted co-server executing inside a secure coprocessor; and enabling clients to interact with the server and the co-server; and

using the trusted co-server as a trusted third party in interactions between the client and the server.

34. (Currently Amended) A trusted co-server, executing a program such that:

for multiple parties, including a Web server a remote client and said co-server, each party may, optionally, provide input, and then the co-server carries out for each party, a function on all these inputs, and optionally returns output to said each party; and

wherein the co-server executes so that parties ~~such as the remote client~~ can authenticate and trust the correct execution of the co-server, in interactions between the client and the server, despite attempts by the Web server to subvert this.

35. (Original) A trusted co-server according to Claim 34, wherein the co-server executes inside a tamper respondent secure coprocessor.

36. (Original) A trusted co-server according to Claim 34, wherein the secure coprocessor is co-located at said server.

37. (Currently Amended) A method of enhancing the security of a Web based transaction utilizing a server, the method comprising the steps:

providing the server with a trusted co-server; and

using the trusted co-server to execute a program such that:

for multiple parties,

each party may, optionally, provide input and then said co-server carries out for each party, a function on all these inputs and the parties trust interactions between the parties and the server.

38. (Original) A method according to claim 37, where:
one party is a Web server and another party is a remote client.

39. (Currently Amended) A method according to Claim 37, where:
the client authenticates the co-server,
the client sends its input to the co-server over a private channel, ~~such as one established by encryption with a shared secret key~~, the co-server sends its output to said another party over a private channel, such as one established by encryption with a shared secret key.

40. (Currently Amended) A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for enhancing a computational service to at least one client of a plurality of clients, said method steps comprising:

moving a selected portion of a computation from a server into a trusted co-server executing inside a secure coprocessor; and

allowing each client to interact with the server and the co-server; and

using the trusted co-server as a trusted third party in interactions between the client and the server.

41. (Original) A program storage device according to Claim 40, wherein the step of allowing includes providing a trusted third party at said server.

42. (Original) A program storage device according to Claim 41, wherein the step of allowing includes enabling said client an authenticated, private channel to said co-server